

В задаче требуется найти минимально возможный правильный RWA и указать правильное распределение на группы, которое позволит получить его. Ввиду отсутствия максимально правильного быстрого решения в нашей команде предпочтение будет отдано решению, наиболее сбалансированному по точности/времени исполнения.

Процесс создания модели

Первоначально была создана программа для случая А, которую планировалось с некоторыми преобразованиями использовать также и для случая В, но было выявлено критическое допущение. Подразумевалась сортировка, а в последствии перебор процентных ставок от минимальной до этой же + 0.15, а следующий перебор шел от первого элемента, не удовлетворяющего прошлому условию. Был выявлен случай ошибки:

Актив	1	1	1	1	1	1	0	0	0
Пассив	0	0	0	0	0	0	1	1	1
Ставка	0	0	0	0.15	0.15	0.15	0.3	0.3	0.3

Случай В:

Алгоритм случая В:

- 1) Необходимо отсортировать входные данные по процентной ставке
- 2) Разделить массив входных данных по процентной ставке на 0,15%
- 3) Отсортировать все ячейки по 0,15% по сроку погашения
- 4) Разделить эти ячейки на группы по 30 дней (группы неттирования)
- 5) Выполняем неттирование

Сложность программы – $O(5n + 2 \text{ сортировки})$, где n – кол-во контрактов в портфеле.

Плюсы алгоритма

К плюсам данного алгоритма можно отнести малое время выполнения для больших данных при малой погрешности, в отличие от перебора, отсутствие ошибки в отрицательную сторону, а значит, что банк не сможет закрыться в случае форс-мажорных обстоятельств, связанных с показателем RWA.

Минусы алгоритма

К минусам данного алгоритма можно отнести наличие незначительных погрешностей по сравнению с перебором.

Почему этот алгоритм?

Мы считаем, что данный алгоритм является наиболее сбалансированным относительно времени и точности исполнения, что позволяет ему считаться оптимальным решением.

Почему не подходит перебор?

Перебор, хоть и позволит получить лучший результат, будет выполняться значительно дольше (сложность перебора приблизительно $O(n^3)$), что сильно увеличит время исполнения программы «в банковских масштабах».

Код программы (на Python 3):

```
k = 0
while i < 2175:
    g = []
    x = s[i][0] + 0.15
    j = i
    net = 0
    k = 0
    o = 0
    while j < 2175 and s[j][0] <= x:
        g.append(s[j][1:] + [s[j][0]])
        j += 1
        o += 1
    g.sort()
    while k < o:
        y = g[k][0] + 30
        while k < o and g[k][0] <= y:
            g[k][5] = z
            net = net + g[k][2] - g[k][3]
            k += 1
        z += 1
    i += k
    if net >= 0:
        active += net
    else:
        passive += abs(net)
    for c in g:
        v.append([c[2], c[3], c[1], c[0], c[6], c[5], c[4]])
rwa = 0.1 * (active + passive) + 0.4 * abs(active - passive)
for i in range(2175):
    l[v[i][6]-1] = v[i]
for i in range(2175):
    if int(l[i][0]) == l[i][0]:
        l[i][0] = int(l[i][0])
    if int(l[i][1]) == l[i][1]:
        l[i][1] = int(l[i][1])
    if int(l[i][3]) == l[i][3]:
        l[i][3] = int(l[i][3])

print(int(round(rwa, 0)))
print(l[i][5])
```